LaTeX Author Guidelines for ICCV Proceedings

Anonymous ICCV submission

Paper ID ****

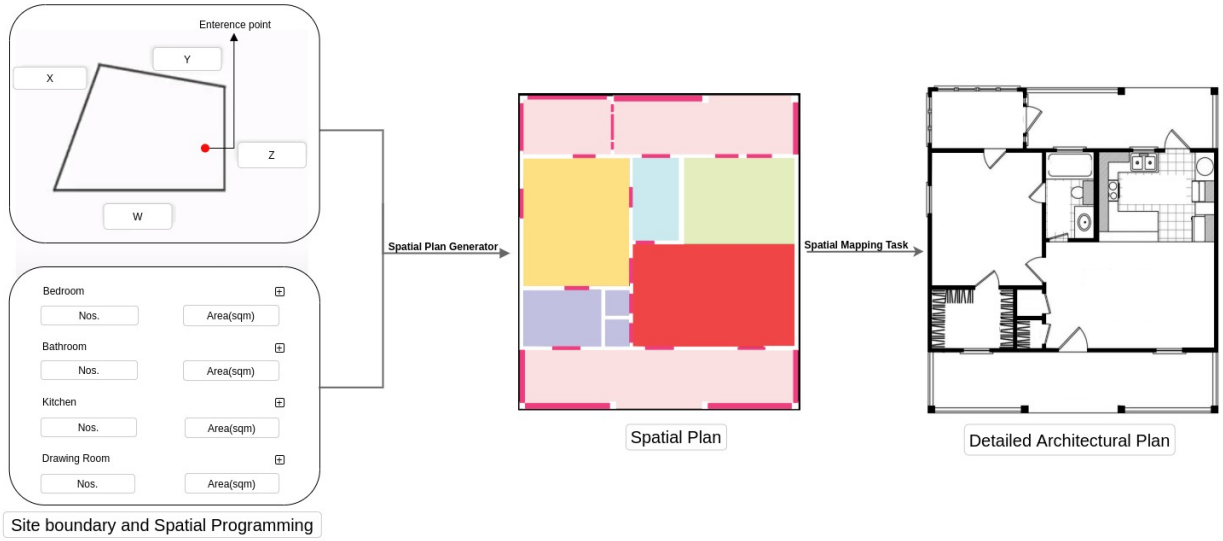


Figure 1. The conceptual visualisation of ArP-Gen consisting of two primary tasks: Spatial Plan Generator(SPG) and Spatial plan Generator(SPT)

Abstract

In architecture, a plan generation is unique problem where the goal is to learn mapping from given site boundaries or building foot print and other basic room requirements. A lot of studies have already been done in the area of generative algorithm to create spatial plan. In this paper, we attempt to propose first end-to-end Architectural plan generation model called ArP-Gen: Architectural Plan Generator, that create architectural plan from basic user inputs like site boundary, entrance point, and room programming and new Spatial Mapping Task (SMT), which maps spatial or semantically segmented floor plans to detail architectural plan, learning detailing and room semantics. Our other contribution includes creation of expert annotated data set for mapping spatial plan to render architectural plan. Experiments demonstrate the potential of ArP-Gen framework to generate feasible Architectural solutions.

1. Introduction

In architecture, a floor plan is a drawing to scale, showing of the relationships between rooms, spaces, traffic patterns, and other physical features. As we know that this process is highly iterative and requires a lot of effort from the designers.

The steps involved in this process can be comprehensively divided into the four major phases. They are as follows.

1. **Programming** (pre-design phase)
2. **Schematic Design**
3. **Design Development**
4. **Construction Drawing**

All the above steps takes lot of time, effort and skill of Architects that will again undergoes few more iteration as per the client requirement.

Roughly, Programming involves decisions related to requirement of rooms and basic design idea. The next two step help in the development of semantically segmented

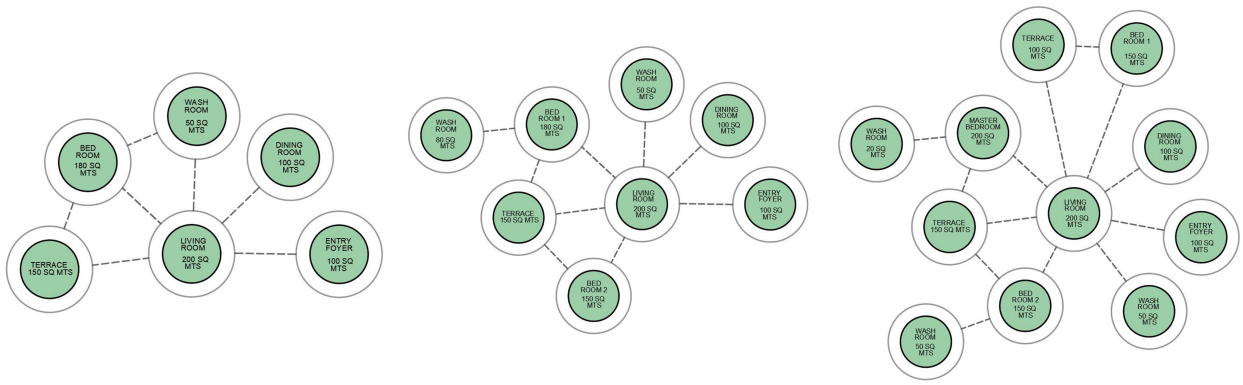


Figure 2. visualization of proximity between the spaces in the form of Bubble diagram. Here the dotted line represented connectivity with spaces. (The bubble diagram of 1 BHK, 2 BHK, and 3 BHK are from left to rigth)

floor plans[1] (Spatial configuration) and last step is Construction of Detail architectural drawing.

Currently, we can find many computer applications which help humans to design floor plans. However, these programs are not completely capable of automating the entire design process. By automating the above process we can save the time and efforts made by the architects in generating iterations. Such automation would allow architects / designers to quickly produce as many floor plans as possible in small span of time.

Spatial configuration is an important area of inquiry related to physical design configuration. It is mostly concerned with finding feasible locations and dimensions for a set of interrelated spaces. Meeting certain mandatory design requirements along with maximizing the quality based on preferences is necessary for a good design. In developing country, more than 70% of the houses in suburban and rural areas are made with little or no involvement of an Architect or Civil Engineer. This is mainly because the hiring an architect can be expensive for the low income people as well as the construction cost involved in such individual housing projects is also very low. Due to limited budget people generally tend to design their own homes and avoid hiring architects or civil engineers. Most of the rough ideas are given by the users and altered executed by local masons who have little to no knowledge of Design principles. Therefore their houses may lack essential design features. Thus, this kind of automated application can even be helpful for them as most of the houses are built on typical design principles.

Also designing a house floor plan is both professional and personal. Many times, people feel the need to remodel or build their home. But due to lack of training these people might lack aptitude to do so. There are also many non-architects/designers who are also looking to remodel houses. As discussed earlier, hiring architects/designers can be quite expensive. Therefore, making the design process hand for common untrained people should be considered. Automation with the help of software or model can be a natural step in this direction. There are many software available in market like Revit, AutoCAD, Sketchup, Rhino, etc which are used to make floor plan layouts but are not intelligent in terms of design. Spatial configuration is an important area of inquiry related to physical design configuration. It is mostly concerned with finding feasible locations and dimensions fora set of interrelated spaces. Meeting certain mandatory design requirements along with maximizing the quality based on preferences is necessary for a good design. Architectural design is particularly interesting because it not only addresses the engineering problems but also deals with optimisation in terms of cost, performance, aesthetics, usability, etc. These abstract qualities are sometimes difficult to describe formally. There have been several researches over past years to automate the process of architectural floor plan layout design with the help of optimisation algorithms along with the help of interactive human decision making.

Also as discussed above, there are many explicit implicit rules of floor plan design. Since the magnitude of variables such as lighting, heating, aesthetic, etc are huge. Generally an architect would need years of architectural training to get expertise in them. Thus, hard coding these rules is a near impossible task. Furthermore this task is highly dependent of conditions of the site and is highly subjective. Thus rather hard coding these rules, it is more desirable that a program could learn these rules from existing examples. Thus the techniques of artificial intelligence and machine Learning will prove to be useful.

Machine Learning methods have been widely adopted in

[1]Semantically segmented floor plan –Each room of the architectural floor plan is rendered with a specific colour code in order to easily differentiate between the spaces.

many fields as a decision-making tool as one can analyse large amounts of data and learn useful features to generate models which can be helpful for prediction and generation. Architectural floor plans consist of elements such as walls, doors, windows, etc and the spatial interrelation in terms of their arrangement and connections. We explored the use of generative evolutionary algorithms along with Generative adversarial network to effectively explore the spatial relations between the floor plans.

The main focus of this research is to make floor plan generators that could utilise the information like the room program with all the areas, all required room connections, entrance, exit, boundary of the site and other important contextual information. Effective application of evolutionary algorithms to generate a huge number of solutions would be the starting point to generate the input for implementation of Conditional Generative Adversarial network to generate detailed layouts.The natural process used by architect involve creating bunch of iterations based on basic architectural principle. Analogously, The Generative algorithm help us to develop iteration of spatial configure plan or semantic segmented plan as they mainly involve heuristic rules. Conditional Generative Adversarial Network [5] has been successfully used to learn image data in pairs and generate new images based on the input. Its power can be utilised to create detailed architectural layouts from the semantically segmented floor plans which are the output of planning with help of evolutionary strategies.

The Main contributions of our paper are as follows:

- We proposed first novel end-to-end architectural plan generator, in which we demonstrate the importance of Genetic algorithm as efficient starting point for Architectural plan generation and Conditional GANs, ability to learn room semantics.
- We also propose new spatial mapping task and introduce Spatial mapping dataset.
- ArP-Gen have shown the ability to produce feasible plans. The code and pre-trained model can be founded here[2]

2. Related work

Spatial plan generation Recently lot of work is done in computer-aided approach for floor plan generation. Existing approaches for floor plan generation include constraint-based systems[10], agent-based systems, evolutionary algorithms[12, 4]. In the approach describe in the paper [12]. The model first placed room which is nearest to entrance point and having highest inter-connectivity among other rooms. After that rooms are placed as per their spatial inter-relational diagram. Generating multiple output by rotating all the spaces iteratively between the boundary condition.

[2]https://github.com/tejasvaidhyadev/ArP-GAN

Figure 3. Visualisation of Annotated data of SMT

Deep Learning for Plan Generation There are only few studies available in the intersection of deep learning and Architectural Plan Generation. One of them uses Conditional GANs [3] to control Design aspect which is one of most important and crucial part in designing.

3. ArP-Gen:Architectural Plan Generator

Mathematically, The problem of Architectural plan generation can be state as follows:

Let the user define parameters as follow:
$b \in$ Site boundary in the form of scale 2D curve
e $\in$ Entrance point in the site plan, and
$f \in$ room program as:

$$f : K \longrightarrow U$$

where $K \in$ Room type and $U \in$ Dimension of room

our aim is learn function $G(b, e, f)$ that will out feasible architectural plan. Interestingly Architectural plan creation is combination of knowledge, experience and context. We define:

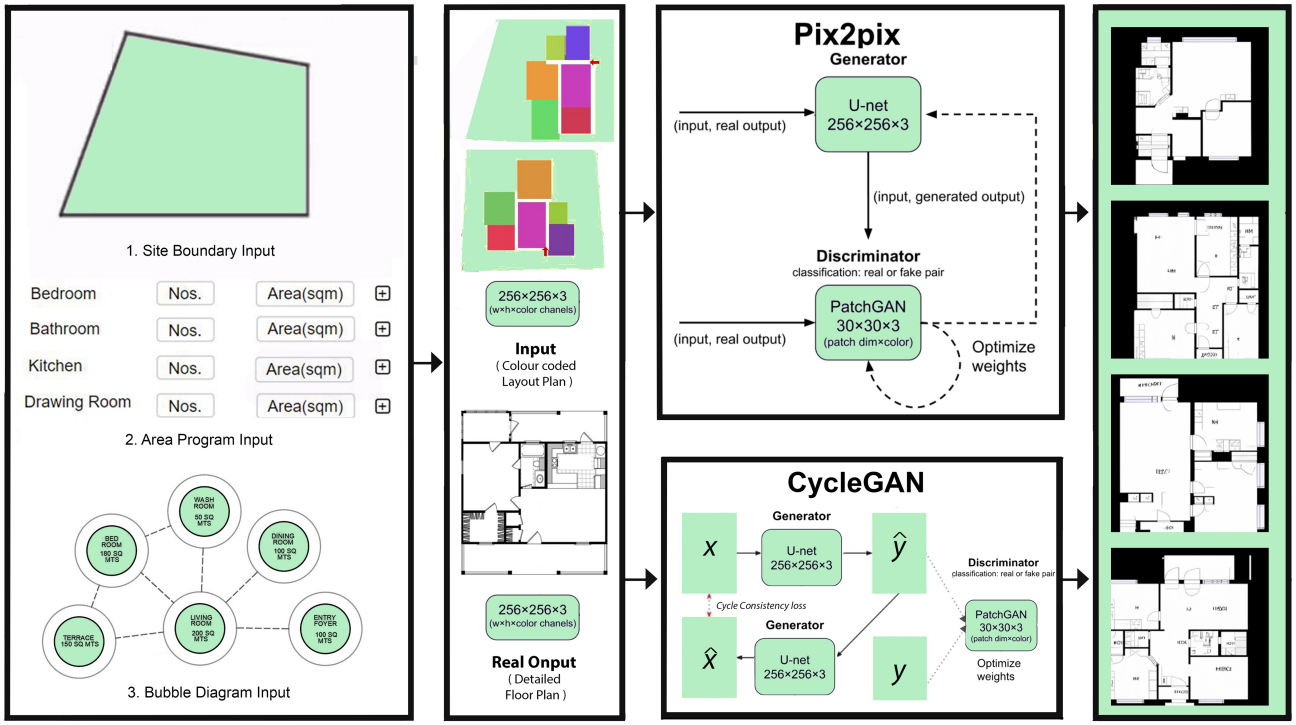
$$G(x) = P(Q(x))$$

$Q \in$ **Spatial Plan Generator** (SPG), that take care of Architectural heuristic rules used in architecture planning inspired from Magnetizing Floor plan [12]. In this paper we only experiment with generation of 1BHK, 2BHK and 3BHK floor plan (in these paper bhk corresponds to bedroom, Hall and Kitchen respectively)[3]

$P \in$ **Spatial Mapping Task** (SMT) helps in learning room semantics and architectural detailing.

The conceptual visualization is shown in figure 1 . The detail description is provided in the section below.

[3]BHK stands for bedroom, hall and kitchen. It is used to convey the number of rooms in a property. For example, a 2BHK means that the particular property has two bedrooms, one hall and a kitchen. A 3BHK unit means there are three bedrooms, a hall and a single kitchen in the property.



(a) Input to Genetic Algorithm (b) Genetic Output (c) Proposed GANs (d) Output

Figure 4. Visualisation of End to end model

Room Type	RGB code
kitchen/hall	RGB(225,255,193)
Bedroom	RGB(254,224,128)
Long Corridor	RGB(255,160,96)
cupboard	RGB (193,254,255)
Toilet	RGB (193,254,255)
terrace and balcony	RGB (254,224,226)
Staircase	RGB(121,63,13)

Table 1. Code codes of annotated Dataset.

	Number of Images
Train	3000
Test	1000
Dev	1000

Table 2. Stats of Dataset relased.

3.1. Spatial Plan Generator

Spatial Plan Generator takes inputs from users(In order to process input b, e, f we need to use emulator, we used mcneel-rhinoceros [9] for our experiments) and generate few iteration of color-coded semantically segmented floor plan.

Our model uses Magnetizing Floor plan Generator modified as per the indian context.

Magnetizing Floor plan : The main inputs of the generator include the room program with all areas as represented by Bubble diagram[4], all required room connections, entrance point and the boundary of the site. It works on a square grid, though the size of a cell can be altered to get more precise or faster results.

We found out that the adjacency matrix for 1BHK, 2BHK and 3BHK plan in india are more or less same. By keeping optimal parameter we can generate iteration of feasible architectural plan. Our finding suggests the spatial plan obtain is good enough for further processing. We further believe that other Geographical factor in the existing genetic algorithmic framework can be easily incorporated making it more architecturally aware. The fixed bubble diagrams for 1BHK, 2BHK and 3BHK are shown in fig. 2

The generator can be easily extended for generating apartments, city or town plans etc.

You can also checkout the hosted demonstration at https://url

[4]Spatial inter-relational diagram

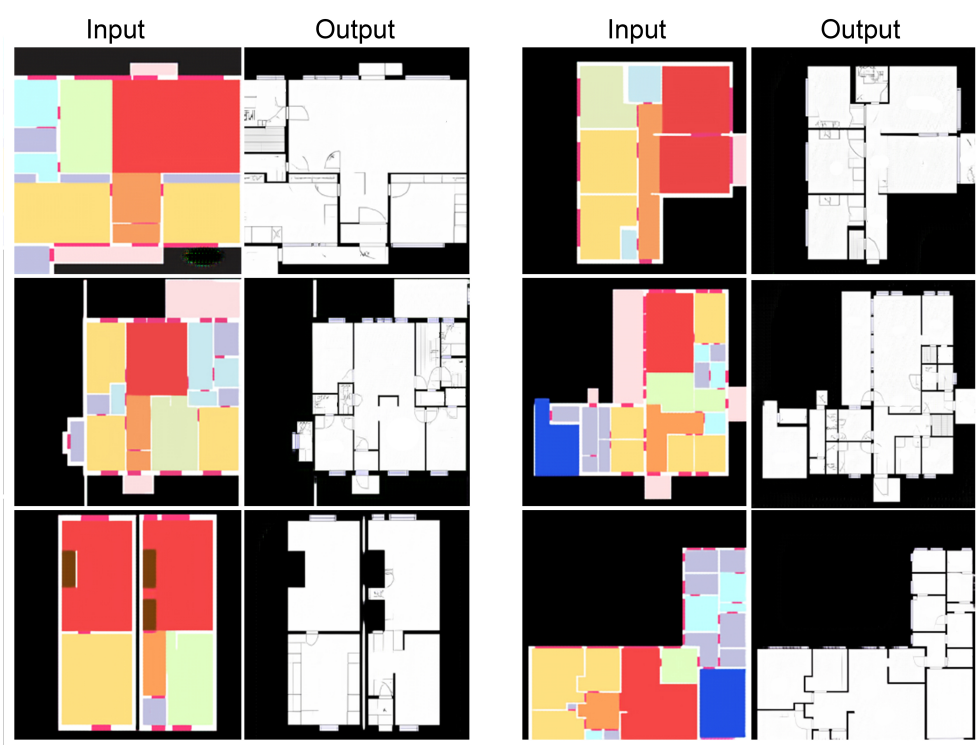


Figure 5. Visualisation of results obtain from SPG (left) and SPT (right) consist of randomly sampled output of both Cyclic and Pix2Pix GANs

3.2. Spatial Mapping Task

In this section, we will introduce novel Image translation or mapping task in the domain of Architecture and Planning. We believed in future we can create model specifically to capture architectural details. In this paper we stick to traditional image translation or style transfer architecture as baselines.

3.2.1 Task Description

The task maps the output $X \in$ SPG to detail architectural plan Y. Our goal is to learn a mapping $G : X ßY$ such that the distribution $G(X)$ is indistinguishable from the distribution Y

Generally, Working with CAD level information for detailing is difficult and on top deeplearning model uses pixel level information. Thus we convert input to pixel level information to easily perform deep learning Algorithm. Deep Learning Algorithm helps to learn room level information in each room. We used conditional GANs for mapping spatial plan to detailed Architectural plan.

We have asked annotators to strictly use the color scheme provided, and hypothesis the GANs will get the reinforced semantic information via color. The dataset generated used to model the mapping from spatial to color-coded plan. The Visualisation of dataset is shown in figure 3

3.2.2 SMT Dataset

We presented the first ever 5K+ expert annotated dataset mapping spatial color-coded plan to architectural detailed plan. We took initial architectural plan from CubiCase5k. The CubiCase5k [7] dataset consists of 5000 floorplans (with man-made annotations) that were collected and reviewed from a larger set of 15,000 mostly Finnish floorplan images.

we removed all the named annotation within the plan images to avoid unnecessary noise and asked annotator consist of architecture students to color code the individual component (such as walls, windows, furniture) as per provided color scheme in Table 1. Each floorplan has been annotated following an annotation protocol which describes the color for annotating the Architectural elements (i.e. windows, doors etc). That is to reinforce all available information from the color-annotated elements (e.g. walls, windows, doors, kitchen etc.) for a given floorplan. The color coding has been done using a adobe photoshop[2] and adobe illustrator [1].

To ensure annotation consistency, each plan is validated by atleast 3 more annotators to verify the color-code scheme as shown in Table 1, 2.O ur dataset consists of variable-resolution images, while our system requires a constant input dimensionality. Therefore, we down-sampled the im-

ages to a fixed resolution of 256 × 256. Given a rectangular image, we first re-scaled the image such that the shorter side was of length 256, and then cropped out the central 256×256 patch from the resulting image.

3.2.3 Models

We used Pix2pix (paired image Translation) and cycleGAN (unpaired image Translation) for our Analysis.

- **Pix2pix**: The Algorithm purposed by Isola P et al [6] is used to learning mapping using paired image.The Pix2Pix model is a type of conditional GAN, or cGAN, where the generation of the output image is conditional on an input, in this case, a source image. The discriminator is provided both with a source image and the target image and must determine whether the target is a plausible transformation of the source image. The generator is trained via adversarial loss, which encourages the generator to generate plausible images in the target domain. The generator is also updated via L1 loss measured between the generated image and the expected output image. This additional loss encourages the generator model to create plausible translations of the source image.

- **CycleGAN**:As described in Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, also known as CycleGAN, a method that can capture the characteristics of one image domain and figure out how these characteristics could be translated into another image domain, all in the absence of any paired training examples. CycleGAN uses a cycle consistency loss to enable training without the need for paired data.

The model is shown in figure 4

3.2.4 Experiments

For the implementation of Cycle GAN, we use pytorch [11] based framework based directly on the model described in the paper [13] that incorporate network containing three convolutions, several residual blocks , two fractionally-strided convolutions with stride 1,2, and one convolution that maps features to RGB. We use 9blocks for 256×256 and higher-resolution training images and instance normalization .

The discriminator networks of both pix2pix and cycleGAN, uses 70 × 70 PatchGANs, which aim to classify whether 70 × 70 overlapping image patches are real or fake [8]. Such a patch-level discriminator architecture has fewer parameters than a full-image discriminator and can work on arbitrarily sized images in a fully convolutional fashion.

4. Results and Inference

The results from SPG and SPT are shown in figure 5 . While we are not making any claim that these samples are better than samples generated by other existing approach, we believe that such end to end frame work highlight the potential of computer vision and genetic algorithm in architectural planning domain.

Limitation:

1. The model still struggle to identify similar architectural details. For instance, windows and doors in SPT Task.

2. In many cases of SPG Task, lots of corridor (SPG) are generating, wasting potential spaces by increasing unnecessary circulation space.

3. Due to resource constraints we only generated limited amount of data and believed by generating more data we can improve the performance of exiting GAN based algorithms

5. Conclusion and Future Works

1. In this paper, We introduce novel application of computer vision (GANs) in Architecture and Regional Planning. We also contributed two tasks and dataset in the community.

2. This paper has demonstrated the viability of the Generation of Architecture plan, suggesting these research directions could prove useful.

For Future work we believe that by generating more dataset, end to end deeplearning model can be explore by replacing genetic algorithm part but keeping bubble diagram as potential input.

6. Acknowledgements

We would like to thank the Uniworks Designs,especially arun chaubey, for their constant support and for providing fund for annotating dataset.

References

[1] Adobe Inc. Adobe illustrator. 5

[2] Adobe Inc. Adobe photoshop. 5

[3] Stanislas Chaillou. Archigan: a generative stack for apartment building design. 3

[4] Egor Gavrilov, Sven Schneider, Martin Dennemark, and Reinhard Koenig. Computer-aided approach to public buildings floor plan generation. magnetizing floor plan generator. *Procedia Manufacturing*, 44:132–139, 01 2020. 3

[5] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. 3

[6] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *CoRR*, abs/1611.07004, 2016. 6

[7] Ahti Kalervo, Juha Ylioinas, Markus Häikiö, Antti Karhu, and Juho Kannala. Cubicasa5k: A dataset and an improved multi-task model for floorplan image analysis, 2019. 5

[8] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation, 2015. 6

[9] Robert McNeel et al. Rhinoceros 3d, version 6.0. *Robert McNeel & Associates, Seattle, WA*, 2010. 4

[10] Pirouz Nourian, Samaneh Rezvani, and Sevil Sariyildiz. Designing with space syntax a configurative approach to architectural layout, proposing a computa- tional methodology. 09 2013. 3

[11] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 6

[12] Sven Schneider, Jan-Ruben Fischer, and Reinhard Koenig. *Rethinking Automated Layout Design: Developing a Creative Evolutionary Design Method for the Layout Problems in Architecture and Urban Design*, pages 367–386. 01 2011. 3

[13] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks, 2020. 6